Average Drift Analysis and Population Scalability

He, Jun; Yao, Xin

Published in:
IEEE Transactions on Evolutionary Computation

DOI:
10.1109/TEVC.2016.2608420

Publication date:
2017

Citation for published version (APA):
He, J., & Yao, X. (2017). Average Drift Analysis and Population Scalability. *IEEE Transactions on Evolutionary Computation*, (99). https://doi.org/10.1109/TEVC.2016.2608420

Average Drift Analysis and Population Scalability

Jun He and Xin Yao

***Abstract*—This paper aims to study how the population size affects the computation time of evolutionary algorithms in a rigorous way. The computation time of evolutionary algorithms can be measured by either the number of generations (hitting time) or the number of fitness evaluations (running time) to find an optimal solution. Population scalability is the ratio of the expected hitting time between a benchmark algorithm and an algorithm using a larger population size. Average drift analysis is introduced to compare the expected hitting time of two algorithms and to estimate lower and upper bounds on the population scalability. Several intuitive beliefs are rigorously analysed. It is proven that (1) using a population sometimes increases rather than decreases the expected hitting time; (2) using a population cannot shorten the expected running time of any elitist evolutionary algorithm on any unimodal function on the time-fitness landscape, however this statement is not true in terms of the distance-based fitness landscape; (3) using a population cannot always reduce the expected running time on deceptive functions, which depends on whether the benchmark algorithm uses elitist selection or random selection.**

***Index Terms*—evolutionary algorithm, computation time, population size, fitness landscape, drift analysis.**

I. Introduction

Population is one of the most important features of evolutionary algorithms (EAs). A wide range of approaches is available to design population-based EAs. Using a population delivers many benefits [1]. The study of the relationship between the performance of an EA and its population size can be traced back to early 1990s. For example, Goldberg et al. [2] presented a population sizing equation to show how a large population size helps an EA to distinguish between good and bad building blocks on some test problems. Mühlenbein and Schlierkamp-Voosen [3] studied the critical (minimal) population size that can guarantee the convergence to the optimum. Arabas et al. [4] proposed an adaptive scheme for controlling the population size, and the effectiveness of the proposed scheme was validated by an empirical study. Eiben et al. [5] reviewed various techniques of parameter controlling for EAs, where the adjustment of population size was considered as an important research issue. Harik et al. [6] linked the population size to the quality of solution by the analogy between one-dimensional random walks and EAs.

The theoretical analysis of the impact of the population size on the computation time of EAs starts in early 2000s [7]. There has been an increasing interest in rigorously analysing the relationship between the computation time of an EA and its population size. The computation time of an EA can be measured by either the expected hitting time or the expected running time. The theoretical studies on this topic can be classified into two directions.

One direction aims to estimate a bound on the computation time of EAs as a function of the population size. This direction belongs to the time complexity analysis of EAs. Drift analysis and tail inequalities are often used for estimating the time bound. This direction may be called a bound-based study. A lot of work has done along this direction. The earliest one was conducted by Jansen et al. [8] who first obtained the cut-off point for a $(1 + \lambda)$ EA on three pseudo-Boolean functions, Leading-Ones, One-Max and Suf-Samp. Jägersküpper and Witt [9] analysed how the running time of a $(\mu+1)$ EA on the Sphere function scales up with respect to μ. Witt [10] proved theoretically that the running time of a $(\mu+1)$ EA on a specific pseudo-Boolean function is polynomial with an overwhelming probability, when μ is large enough. Storch [11] presented a rigorous runtime analysis of the choice of the population size with respect to a $(\mu + 1)$ EA on several pseudo-Boolean functions. Yu and Zhou [12] investigated the expected hitting time of $(\lambda + \lambda)$ EAs when $\lambda = 1$ and $\lambda = n$ (where n is the problem input size) on the trap problem. Oliveto et al. [13] presented a runtime analysis of both $(1 + \lambda)$ and $(\mu+1)$ EAs on some instances of the vertex covering problem. Friedrich et al. [14] analysed the running time of a $(\mu + 1)$ EA with diversity-preserving mechanisms on the Two-Max problem. Chen et al. [15] obtained an upper bound on the hitting time of $(\lambda + \lambda)$ EAs on Leading-Ones and One-Max problems. Lässig and Sudholt [16] presented a running time analysis of a $(1 + \lambda)$ EA with an adaptive offspring size λ on several pseudo-Boolean functions. Rowe and Sudholt [17] discussed the running time of $(1 + \lambda)$ EAs in terms of the offspring population size on unimodal functions. Doerr and Künnemann [18] analysed the time bound of $(1 + \lambda)$ EAs for optimizing linear pseudo-Boolean functions. Doerr and Künnemann [19] showed that $(1+\lambda)$ EAs with even very large offspring populations does not reduce the runtime significantly on the Royal Road function. Oliveto and Witt [20] presented a rigorous running time analysis of the well-known Simple Genetic Algorithm for One-Max. Gießen and Witt [21] studied the relation between the population size and mutation strength for a $(1 + \lambda)$ EA on One-Max.

Another direction aims to calculate a ratio, named popula-

Manuscript received xx xx xxxx

This work was supported by the EPSRC under Grant Nos. EP/I009809/1, EP/I010297/1 and EP/K001523/1, and by NSFC under Grant No. 61329302. XY was also supported by a Royal Society Wolfson Research Merit Award.

Jun He is with Department of Computer Science, Aberystwyth University, Aberystwyth, SY23 3DB, UK. Email: jun.he@aber.ac.uk

Xin Yao is with CERCIA, School of Computer Science, University of Birmingham, Birmingham B15 2TT, UK. Email: x.yao@cs.bham.ac.uk

This article has been accepted for publication in a future issue of IEEE Transactions on Evolutionary Computation, but has not been fully edited. Content may change prior to final publication.

tion scalability, which is described as follows:

$$\frac{\text{expected hitting time of a benchmark EA}}{\text{expected hitting time of an EA using a population}}. \quad (1)$$

This direction may be called a ratio-based study. As one of the earliest analyses, He and Yao [7] investigated how the population scalability of EAs varies as the population size changes on two simple functions (One-Max and Deceptive). In that paper, EAs are assumed to be run on a hypothetical parallel computer, that is, to assign each individual on one processor. If the communication cost is ignored, the population scalability is equivalent to the speedup in parallel computation. The link between population scalability and parallelism was further discussed in [22]. However, since calculating the population scalability is not an easy job, no further development has been made since then.

This paper belongs to the ratio-based study. It is significantly different from the bound-based study. The bound-based study focuses on an asymptotic time bound as a function of the population size. It does not calculate the population scalability and will not answer whether a $(2+2)$ EA is faster than a $(1+1)$ EA. The ratio-based study aims to calculate the population scalability and will answer whether a $(2+2)$ EA is faster than a $(1+1)$ EA. But it is not necessary to estimate an asymptotic time bound.

Compared with previous work on the analysis of population-based EAs, the current paper has two novelties.

1) Average drift analysis is presented as a tool of comparing the expected hitting time of two EAs and studying population scalability. The approach used in [7] is based on the fundamental matrix of absorbing Markov chains. It is hard to calculate the expected hitting time through the fundamental matrix. But using average drift analysis, it is possible to estimate population scalability without calculating the expected hitting time. This is an important improvement in the analysis tool.
2) The scalability threshold replaces the cut-off point. The population threshold is the minimal population size at which the running time of an EA using a larger population size is greater than that of the benchmark EA. The cut-off point [8] is the maximize population size at which the running time of the EA is in the same order as that of the benchmark EA. Let's show the advantage of population scalability by an example: $(1+\lambda)$ EAs (using bitwise mutation and elitist selection) for solving the One-Max problem. According to [8], the cut-off point is $\Theta\Big(\frac{(\ln n)(\ln\ln n)}{\ln\ln\ln n}\Big)$. This means when the population size λ is smaller than the cut-off point, the running time of the $(1+\lambda)$ EA is in the same order as that of the $(1+1)$ EA but different by a constant factor. The constant could be $=1$, >1 or <1. Therefore the cut-off point does not answer the question whether the expected running time of a $(1+\lambda)$ EA (where $2 \leq \lambda = O\Big(\frac{(\ln n)(\ln\ln n)}{\ln\ln\ln n}\Big)$) is smaller or larger than that of the $(1+1)$ EA. However, according to Proposition 4 and its discussion in this paper, the scalability threshold is 2. This means that the running time of the $(1+\lambda)$ EA (for any $\lambda \geq 2$) is larger than that of the $(1+1)$ EA. Therefore the scalability threshold is more accurate than the cut-off point.

With the help of average drift analysis, this paper analyses the following intuitive beliefs in a rigorously way.

1) Using a population "always" reduces the expected hitting time (not running time) of an EA to find an optimal point.
2) Using a population cannot shorten the expected running time of an elitist EA on "unimodal" functions.
3) Using a population can reduce the expected running time of an EA on "deceptive" functions.

The paper is organised as follows: Section II defines population scalability. Section III presents drift analysis for population scalability. Section IV analyses scenario 1: using a population does not reduce the hitting time. Section V analyses scenario 2: using a population reduces the hitting time, but not the running time. Section VI investigates scenario 3: using a population reduces the running time. Section VIII concludes the paper.

II. Population scalability

A. Evolutionary algorithms

Consider the problem of maximizing a function $f(x)$ where $x \in \mathcal{S}$ and $\mathcal{S}$ is a finite set. A point in $\mathcal{S}$ is called a solution or an individual. A population consists of one or more individuals. A $(\mu+\lambda)$ EA is described in Algorithm 1. The stopping criterion is that the EA halts once an optimal solution is found. The criterion is used for the sake of analysis because our interest is the first hitting time (when the EA finds an optimal solution for the first time). If Φ_t includes an optimal solution, assign $\Phi_t = \Phi_{t+1} = \Phi_{t+2} = \cdots$ for ever.

Algorithm 1 A $(\mu+\lambda)$ EA where $\mu, \lambda \geq 1$

1: initialise a population Φ_0 consisting of μ individuals (solutions) and $t \leftarrow 0$;
2: evaluate the fitness of individuals in Φ_0;
3: **while** Φ_t does not include an optimal solution **do**
4: mutate (or crossover) individuals in Φ_t and generate a children population Ψ_t consisting of λ individuals;
5: evaluate the fitness of individuals in Ψ_t;
6: probabilistically select μ individuals from $\Phi_t \cup \Psi_t$ as Φ_{t+1};
7: $t \leftarrow t+1$;
8: **end while**

The sequence $\{\Phi_t; t = 0, 1, \cdots\}$ can be modelled by a Markov chain [23]. Each generation of the EA consists of two steps: to generate new individuals by mutation or crossover and to select individuals for next generation,

$$\Phi_t \overset{\text{mutation (or crossover)}}{\longrightarrow} \Psi_t \cup \Phi_t \overset{\text{selection}}{\longrightarrow} \Phi_{t+1}.$$

Let $\mathcal{P}$ denote the set of all populations, $\mathcal{P}_{\text{opt}}$ the set of populations including an optimal solution and $\mathcal{P}_{\text{non}}$ the set of populations without an optimal solution. The transition from Φ_t to Ψ_t can be represented using mutation (or crossover) probabilities:

$$P_{\text{m}}(X, Y) \overset{\text{def}}{=} P(\Psi_t = Y \mid \Phi_t = X), \quad X, Y \in \mathcal{P}, \quad (2)$$

where Φ_t, Ψ_t are random variables representing the tth generation population and its children population. X, Y are their values taken from $\mathcal{P}$.

The transition from Φ_t and Ψ_t to Φ_{t+1} can be represented using selection probabilities:

$$P_{\mathrm{s}}(X, Y, Z) \stackrel{\text{def}}{=} P(\Phi_{t+1} = Z \mid \Phi_t = X, \Psi_t = Y). \quad (3)$$

The transition from Φ_t from Φ_{t+1} can be represented using transition probabilities:

$$P(X, Y) \stackrel{\text{def}}{=} \Pr(\Phi_{t+1} = Y \mid \Phi_t = X). \quad (4)$$

The hitting time is the number of generations of an EA to find an optimal solution for the first time.

Definition 1: Given $\Phi_0 = X$, the *expected hitting time* of an EA is defined by

$$g(X) \stackrel{\text{def}}{=} \sum_{t=0}^{+\infty} \Pr(\Phi_t \in \mathcal{P}_{\mathrm{non}}). \quad (5)$$

If the initial population Φ_0 is chosen according to a probability distribution over $\mathcal{P}$, the expected hitting time is given by

$$g(\Phi_0) \stackrel{\text{def}}{=} \sum_{X \in \mathcal{P}} g(X) \Pr(\Phi_0 = X).$$

The expected running time of a $(\mu + \lambda)$ EA is the expected number of fitness evaluations, which equals to $\mu + \lambda g(\Phi_0)$. For the sake of simplicity, we always omit the first term μ, which is the number of fitness evaluations in initialization.

If genetic operators don't change in time, the sequence $\{\Phi_t; t = 0, 1, \cdots\}$ can be modelled by a homogeneous Markov chain. According to the fundamental matrix theorem [24, Theorem 11.5], the expected hitting time of an EA can be calculated from transition probabilities.

Theorem 1: If the population sequence $\{\Phi_t, t = 0, 1, \cdots\}$ is a homogeneous Markov chain and converges to $\mathcal{P}_{\mathrm{opt}}$, that is, $\lim_{t \to +\infty} \Pr(\Phi_t \in \mathcal{P}_{\mathrm{opt}}) = 1$, then the expected hitting time $g(X)$ satisfies a linear equation system:

$$\begin{cases} g(X) = 0, & \text{if } X \in \mathcal{P}_{\mathrm{opt}}, \\ \sum_{Y \in \mathcal{P}} P(X, Y)\,(g(X) - g(Y)) = 1, & \text{if } X \notin \mathcal{P}_{\mathrm{non}}. \end{cases} \quad (6)$$

The fundamental matrix theorem is useful in analysing elitist EAs [7], [23], [25]. However, its disadvantage is the difficulty of solving the above linear equation system.

B. Population scalability

The population scalability is defined as the ratio of the expected hitting time between a benchmark EA and an EA with a larger population size. In this paper, the benchmark is a $(1+1)$ EA using mutation and selection operators. Other types of EAs may play the role of a benchmark too. For example, a $(2+2)$ EA could be chosen as a benchmark when studying EAs with crossover. But we will not discuss them here.

Definition 2: Given a $(1 + 1)$ EA and a $(\mu + \lambda)$ EA that exploit an identical mutation operator to optimise the same fitness function, let $\Phi_0^{(1+1)}$ and $\Phi_0^{(\mu+\lambda)}$ denote their corresponding initial populations, then the *population scalability* is defined by

$$PS(\mu + \lambda \mid \Phi_0^{(1+1)}, \Phi_0^{(\mu+\lambda)}) \stackrel{\text{def}}{=} \frac{g^{(1+1)}(\Phi_0^{(1+1)})}{g^{(\mu+\lambda)}(\Phi_0^{(\mu+\lambda)})}, \quad (7)$$

where the superscripts $^{(1+1)}$ and $^{(\mu+\lambda)}$ are used to distinguish the $(1 + 1)$ EA and $(\mu + \lambda)$ EA.

An essential part of the definition above is that both EAs must adopt identical mutation operators. This ensures that the comparison is meaningful. Nonetheless, it is impossible for the selection operators to be identical. Indeed even if the selection operators are of the same type, for example roulette wheel selection, the conditional probabilities determining the actual selection operators are never identical under distinct population sizes.

Obviously the value of population scalability relies on initial populations. Due to the use of a population, $\Phi_0^{(\mu+\lambda)}$ may contain several individuals some of which are different from $\Phi_0^{(1+1)}$. For the sake of comparison, we restrict our discussion to identical initialization, that is, for the $(1+1)$ EA, $\Phi_0^{(1+1)} = x$ and for the $(\mu + \lambda)$ EA, $\Phi_0^{(\mu+\lambda)} = (x, \cdots, x)$. In this case, $PS(\mu + \lambda \mid \Phi_0^{(1+1)}, \Phi_0^{(\mu+\lambda)})$ is denoted by $PS(\mu+\lambda \mid x)$ in short. There exist other types of initialization but we will not discussed them here.

The notion of population scalability is similar to that of the speedup widely used in analyzing parallel algorithms. The speedup of parallel EAs have been studied through experiments [26], [27], [28]. If each individual is assigned to a processor, then EAs turn into parallel EAs. Under this circumstance, population scalability is equivalent to speedup if ignoring the communication cost. Hence population scalability is called speedup on a hypothetical parallel computer in [7].

The following questions are essential when studying population scalability.

1) Given a $\lambda \geq 2$ or $\mu \geq 2$, is the population scalability $PS(\mu + \lambda \mid x) > 1$?
   If it is, we may assign each individual to a processor in a parallel computing system and then the CPU computation time of the $(\mu + \lambda)$ EA is less than that of the $(1 + 1)$ EA (if ignoring the communication cost).
2) Given a $\lambda \geq 2$ or $\mu \geq 2$, is the population scalability $PS(\mu + \lambda \mid x) > \lambda$?
   If it is, then the CPU computation time of the $(\mu + \lambda)$ EA on a computer is less than that of the $(1 + 1)$ EA.
3) Where are the smallest population sizes (μ, λ) such that the expected running time of the $(\mu + \lambda)$ EA is larger than that of the $(1 + 1)$ EA?
   We call this point the scalability threshold, which satisfies

$$\begin{cases} \min\{\mu : PS(\mu + \lambda \mid x) > \lambda\}, \\ \min\{\lambda : PS(\mu + \lambda \mid x) > \lambda\}. \end{cases} \quad (8)$$

   In general, the scalability threshold is not a single point but a Pareto front due to minimizing both population sizes μ and λ simultaneously. However, in a $(1 + \lambda)$ or $(\lambda + \lambda)$ EA, the scalability threshold is a single point.

III. Average drift analysis and time-fitness landscape

A. Average drift analysis

Average drift analysis is a variant of drift analysis for estimating the expected hitting time of EAs. The idea of average drift was first used by Jägersküpper who considered the average drift of a $(1+1)$ EA on linear functions and provided a delicate analysis of the running time of the $(1+1)$ EA [29]. Nevertheless his work was restricted to the $(1+1)$ EA and linear functions. Even the term of average drift did not appear in [29] but was first adopted by Doerr [30] for introducing Jägersküpper's work [29]. In this section, the average drift is formally defined and then general average drift theorems are presented.

In drift analysis, a distance function $d(X)$ is used to measure how far a population X is away from the optimal set $\mathcal{P}_{\rm opt}$. It is a non-negative function such that $0 < d(X) < +\infty$ for any $X \in \mathcal{P}_{\rm non}$ and $d(X) = 0$ for $X \in \mathcal{P}_{\rm opt}$. Drift is used to measure the progress rate of a population moving towards the optimal set per generation.

Definition 3: Given a population X, the *pointwise drift* at X is

$$\Delta(X) \stackrel{\rm def}{=} \sum_{Y\in\mathcal{P}} (d(X)-d(Y))\Pr(\Phi_{t+1}=Y \mid \Phi_t = X). \quad (9)$$

Given a generation t, the *average drift* at t is

$$\bar{\Delta}_t \stackrel{\rm def}{=} \begin{cases} 0, & \text{if } \Pr(\Phi_t\in\mathcal{P}_{\rm non})=0, \\ \frac{\sum_{X\in\mathcal{P}_{\rm non}} \Delta(X)\Pr(\Phi_t=X)}{\Pr(\Phi_t\in\mathcal{P}_{\rm non})}, & \text{otherwise.} \end{cases} \quad (10)$$

The following theorem provides an approach to estimating a lower bound on the expected hitting time. It is a variation of [31, Theorem 4].

Theorem 2: Provided that the population sequence $\{\Phi_t, t = 0,1,\cdots\}$ converges to $\mathcal{P}_{\rm opt}$ where Φ_0 satisfies $\Pr(\Phi_0 \in \mathcal{P}_{\rm non}) > 0$. Given a distance function $d(X)$, if for any t and any Φ_t such that $\Pr(\Phi_t \in \mathcal{P}_{\rm non}) > 0$, the average drift $\bar{\Delta}_t \le c$ where $c > 0$, then the expected hitting time $g(\Phi_0) \ge d(\Phi_0)/c$, where

$$d(\Phi_0) \stackrel{\rm def}{=} \sum_{X\in\mathcal{P}} d(X)\Pr(\Phi_0 = X).$$

Furthermore if for at least one t, the average drift $\bar{\Delta}_t < c$, then $g(\Phi_0) > d(\Phi_0)/c$.

Proof: Without loss of generality, let $c = 1$. From the condition $\bar{\Delta}_t \le 1$ for any t and any Φ_t such that $\Pr(\Phi_t \in \mathcal{P}_{\rm non}) > 0$, we have

$$\begin{aligned} &\Pr(\Phi_t \in \mathcal{P}_{\rm non}) \\ &\ge \sum_{X\in\mathcal{P}_{\rm non}} \Delta(X)\Pr(\Phi_t = X) \\ &\ge \textstyle\sum_{X\in\mathcal{P}_{\rm non}} d(X)\Pr(\Phi_t=X) - \sum_{Y\in\mathcal{P}_{\rm non}} d(Y)\Pr(\Phi_{t+1}=Y). \end{aligned} \quad (11)$$

Summing the term $\Pr(\Phi_t \in \mathcal{P}_{\rm non})$ from $t = 0$ to k, we get

$$\begin{aligned} &\sum_{t=0}^{k} \Pr(\Phi_t \in \mathcal{P}_{\rm non}) \\ &\ge \textstyle\sum_{t=0}^{k}(\sum_{X\in\mathcal{P}_{\rm non}} d(X)\Pr(\Phi_t=X) - \sum_{Y\in\mathcal{P}_{\rm non}} d(Y)\Pr(\Phi_{t+1}=Y)) \\ &= \textstyle\sum_{X\in\mathcal{P}_{\rm non}} d(X)\Pr(\Phi_0=X) - \sum_{Y\in\mathcal{P}_{\rm non}} d(Y)\Pr(\Phi_{k+1}=Y). \end{aligned} \quad (12)$$

Notice that

$$\begin{aligned} &\textstyle\sum_{Y\in\mathcal{P}_{\rm non}} d(Y)\Pr(\Phi_{k+1}=Y) \\ &\le \max_{X\in\mathcal{P}} d(X) \textstyle\sum_{Y\in\mathcal{P}_{\rm non}} \Pr(\Phi_{k+1}=Y) \\ &= \max_{X\in\mathcal{P}} d(X)\Pr(\Phi_{k+1}\in\mathcal{P}_{\rm non}). \end{aligned} \quad (13)$$

Since the EA is convergent: $\lim_{k\to+\infty}\Pr(\Phi_{k+1}\in\mathcal{P}_{\rm non}) = 0$, then from Inequality (13) we have

$$\lim_{k\to+\infty} \sum_{Y\in\mathcal{P}_{\rm non}} d(Y)\Pr(\Phi_{k+1}=Y) = 0. \quad (14)$$

Applying the above result to Inequality (12) (let $k \to +\infty$), we get

$$\begin{aligned} g(\Phi_0) &= \sum_{t=0}^{+\infty} \Pr(\Phi_t \in \mathcal{P}_{\rm non}) \\ &\ge \sum_{X\in\mathcal{P}_{\rm non}} d(X)\Pr(\Phi_0 = X) = d(\Phi_0), \end{aligned} \quad (15)$$

which gives the desired result. If for some t, the average drift $\bar{\Delta}_t < 1$, then Inequality (12) is strict for any $k \ge t$ and Inequality (15) is strict too. ∎

Similarly, the theorem below provides an approach to estimating an upper bound on the expected hitting time. It is a variation of [31, Theorem 1]. Its proof is similar to that of the above theorem.

Theorem 3: Provided that population sequence $\{\Phi_t, t = 0,1,\cdots\}$ converges to $\mathcal{P}_{\rm opt}$ where Φ_0 satisfies $\Pr(\Phi_0 \in \mathcal{P}_{\rm non}) > 0$. Given a distance function $d(X)$, if for any t and any Φ_t such that $\Pr(\Phi_t \in \mathcal{P}_{\rm non}) > 0$, the average drift $\bar{\Delta}_t \ge c$ where $c > 0$, then the expected hitting time $g(\Phi_0) \le d(\Phi_0)/c$. Furthermore if for at least one t, the average drift $\bar{\Delta}_t > c$, then $g(\Phi_0) < d(\Phi_0)/c$.

Pointwise drift theorems are corollaries of average drift theorems, because it requires a stronger condition on the pointwise drift: $\Delta(X) \ge c$ (or $\le c$) for any $X \in \mathcal{P}_{\rm non}$. It implies the average drift $\bar{\Delta}_t \ge c$ (or $\le c$) for any $\Phi_0 \in \mathcal{P}_{\rm non}$.

Theorem 4: [23, Theorem 2] Provided that population sequence $\{\Phi_t, t = 0,1,\cdots\}$ converges to $\mathcal{P}_{\rm opt}$. Given a distance function $d(X)$, if for any $X \in \mathcal{P}_{\rm non}$, the pointwise drift $\Delta(X) \le c$ (where $c > 0$), then for any initial population $X_0 \in \mathcal{P}_{\rm non}$, the expected hitting time $g(X_0) \ge d(X_0)/c$.

Theorem 5: [23, Theorems 3] Provided that population sequence $\{\Phi_t, t = 0,1,\cdots\}$ converges to $\mathcal{P}_{\rm opt}$. Given a distance function $d(X)$, if for any $X \in \mathcal{P}_{\rm non}$, the pointwise drift $\Delta(X) \ge c$ where $c > 0$, then for any initial population $X_0 \in \mathcal{P}_{\rm non}$, the expected hitting time $g(X_0) \le d(X_0)/c$.

The average drift analysis provides a useful tool for comparing the expected hitting time of two EAs. Its idea is simple. One EA is taken as the benchmark and its expected hitting time is used to define a distance function for the other EA. Then the average drift of the other EA is estimated and then its expected hitting time is bounded using average drift analysis.

Theorem 6: Given two EAs A and B to optimise the same fitness function, let $\{X_t, t = 0,1\cdots\}$ and $\{Y_t, t = 0,1,\cdots\}$ denote their population sequences respectively. For algorithm B, define a distance function $d^B(X)$ such that $d^B(Y_0) = g^A(X_0)$, where $g^A(X_0)$ is the expected hitting time of algorithm A starting at X_0. If for any t and any Y_t such

that $\Pr(Y_t \in \mathcal{P}_{\text{non}}) > 0$, average drift $\bar{\Delta}_t^B \leq c$ where $c > 0$, then the expected hitting time of algorithm B satisfies that $g^B(Y_0) \geq g^A(X_0)/c$. Furthermore if for at least one $t \geq 0$, $\bar{\Delta}_t^B < c$, then $g^B(Y_0) > g^A(X_0)/c$.

Proof: It is a direct corollary of Theorem 2. ■

Theorem 7: Given two EAs A and B to optimise the same fitness function, let $\{X_t, t = 0, 1 \cdots\}$ and $\{Y_t, t = 0, 1, \cdots\}$ be their population sequences respectively. For algorithm B, define a distance function $d^B(X)$ such that $d^B(Y_0) = g^A(X_0)$, where $g^A(X_0)$ is the expected hitting time of algorithm A starting at X_0. If for any t and any Y_t such that $\Pr(Y_t \in \mathcal{P}_{\text{non}}) > 0$, average drift $\bar{\Delta}_t^B \geq c$ where $c > 0$, then $g^B(Y_0) \leq g^A(X_0)/c$. Furthermore if for some $t \geq 0$, $\bar{\Delta}_t^B > c$, then $g^B(Y_0) < g^A(X_0)/c$.

Proof: It is a direct corollary of Theorem 3. ■

B. Average drift analysis for population scalability

Average drift analysis for estimating the population scalability is based on an simple idea. Given a benchmark EA and another EA using a larger population size, we assume that both EAs start at the same point with an equal distance to the optimal set. If at each generation, the average drift of the other EA is 10 times that of the benchmark EA, then the expected hitting time of the other EA will be $1/10$ of that of the benchmark EA. Thus the population scalability is 10. This simple idea can be formalised as follows.

Consider a $(1+1)$ EA and a $(\mu+\lambda)$ EA (where $\lambda \geq 2$) that exploit an identical mutation operator to optimise the same fitness function. Provided that $\Phi_0^{(1+1)} = x_0$ and $\Phi_0^{(\mu+\lambda)} = (x_0, \cdots, x_0)$ for some $x_0 \in \mathcal{P}_{\text{non}}$, for the $(\mu + \lambda)$ EA, define a distance function $d(X)$ such that $d^{(\mu+\lambda)}(x_0, \cdots, x_0) = g^{(1+1)}(x_0)$.

The first theorem establishes a sufficient condition for estimating the upper bound on population scalability. Thanks to average drift analysis, there is no requirement that the population sequence is a Markov chain.

Theorem 8: If for all $t \geq 0$ and any $\Phi_t^{(\mu+\lambda)}$ such that $\Pr(\Phi_t^{(\mu+\lambda)} \in \mathcal{P}_{\text{non}}) > 0$, average drift $\bar{\Delta}_t^{(\mu+\lambda)} \leq c$ where $c > 0$, then $PS(\mu + \lambda \mid x_0) \leq c$. Furthermore if for at least one $t \geq 0$, average drift $\bar{\Delta}_t^{(\mu+\lambda)} < c$, then $PS(\mu+\lambda \mid x_0) < c$.

Proof: According to Theorem 6, the expected hitting time satisfies: $g^{(\mu+\lambda)}(x_0) \geq g^{(1+1)}(x_0)/c$. Then we have $PS(\mu + \lambda \mid x_0) \leq c$. If for at least one $t \geq 0$, average drift $\bar{\Delta}_t^{(\mu+\lambda)} < c$, then according to Theorem 6, $g^{(\mu+\lambda)}(x_0) \geq g^{(1+1)}(x_0)/c$ and then $PS(\mu + \lambda \mid x_0) < c$. ■

Similarly, the second theorem establishes a sufficient condition for estimating the lower bound on population scalability. Its proof is the almost the same as the that of the above theorem except using Theorem 7.

Theorem 9: If for all $t \geq 0$ and any $\Phi_t^{(\mu+\lambda)}$ such that $\Pr(\Phi_t^{(\mu+\lambda)} \in \mathcal{P}_{\text{non}}) > 0$, average drift $\bar{\Delta}_t^{(\mu+\lambda)} \geq c$ where $c > 0$, then $PS(\mu + \lambda \mid x_0) \geq c$. Furthermore if for at least one $t \geq 0$, average drift $\bar{\Delta}_t^{(\mu+\lambda)} > c$, then $PS(\mu+\lambda \mid x_0) > c$.

C. Time-fitness landscape

In this paper, unimodal and multi-modal functions are established upon the time-fitness landscape, a concept introduced in [32]. It aims at describing the fitness landscape related to a general search space, which is a finite set.

Definition 4: Given a $(1 + 1)$ elitist EA for maximizing a function $f(x)$, its *time-fitness landscape* is the set of pairs $(g(x), f(x))$, where $g(x)$ is the expected hitting time of the $(1 + 1)$ EA starting at x. The neighbour of x includes two points: the point y_1 such that $g(y_1)$ is the closest to $g(x)$ from the direction $g(y) < g(x)$, and the point y_2 such that $g(y_2)$ is the closest to $g(x)$ from the direction $g(y) > g(x)$.

The time-fitness landscape is completely different from traditional ones based on a distance. The former is related to a $(1 + 1)$ EA, but the latter usually not. Let's show the difference by unimodal functions. A function is called unimodal if every non-optimal point has a neighbour with a strictly better fitness [17]. Traditionally the definition of the neighbour relies on a distance. For example, if the search space is the set of binary strings, the neighbour of a point x includes all points y with Hamming distance 1 from x [17]. But such a definition is applicable to a finite set because the distance is unknown or not defined. Therefore for a general finite set, unimodal functions are defined on the time-fitness landscape instead [32].

Definition 5: Let $\mathcal{S} = \{s_0, s_1, \cdots, s_K\}$ and f a fitness function such that $f(s_0) > f(s_1) > \cdots > f(s_K)$. Given a $(1 + 1)$ elitist EA to maximise a function f, f is called *unimodal* to the $(1+1)$ EA if $g^{(1+1)}(s_1) < \cdots < g^{(1+1)}(s_K)$. A unimodal time-fitness landscape is visualised in Fig. 1.

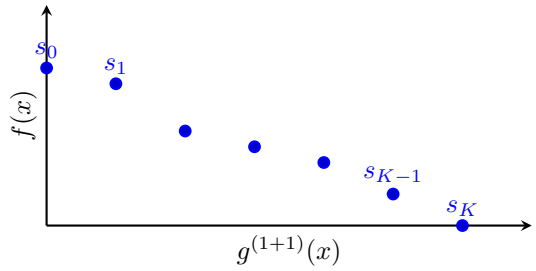

Fig. 1. A unimodal time-fitness landscape. The x axis is the expected hitting time of the $(1 + 1)$ EA. The y axis is the fitness function.

According to [32], unimodal functions to a $(1 + 1)$ elitist EA are the easiest among all fitness functions with the same optimum at s_0. Furthermore, according to [32], for any fitness function $f(x)$, we can construct a $(1 + 1)$ elitist EA to which f is the easiest (unimodal). Therefore any fitness function can be unimodal to a $(1 + 1)$ elitist EA.

Here are two instances of unimodal functions. The TwoMax function [32] below is unimodal under Hamming distance.

$$f(x) = \max\{|x|, n - |x|\}, \quad x \in \{0, 1\}^n, \tag{16}$$

where $|x|$ denotes the number of 1-valued bits. It is unimodal to the following $(1 + 1)$ elitist EA with $p = \frac{1}{n}$ [32].

- **Bitwise Mutation** p**.** Flip each bit independently with flipping probability p.
- **Elitist Selection.** Select the best individual from $\Phi_t \cup \Psi_t$.

The Two Needles in the Haystack function [32] below is also unimodal to the above $(1 + 1)$ elitist EA with $p = \frac{1}{2}$,

although it is regarded as a plateau function under Hamming distance.

$$f(x) = \begin{cases} 1, & \text{if } |x| = 0 \text{ or } |x| = n; \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

IV. Scenario 1: Using a population cannot reduce hitting time

A. Case study 1: two-paths-I functions

It is an intuitive belief that using a population will reduce the number of generations to find an optimal solution. The following case study shows this belief is not always true.

Before the case study, the concept of path [33] is revisited. Given a $(1+1)$ EA for maximizing $f(x)$, a path is a sequence of points $\{x_0 \to x_1 \to \cdots \to x_k\}$ such that $P(x_{i-1}, x_i) > 0$ for $i = 1, \cdots, k$. The path is denoted by $\text{Path}(x_0, x_1, \cdots, x_k)$. The case study is about two-paths-I functions which are defined as below.

Definition 6: Let $\mathcal{S} = \{s_0, s_1, \cdots, s_{K+L}, s_{K+L+1}\}$ where $L > K$ and f a fitness function such that

$$\begin{aligned} &f(s_0) > f(s_{K+1}) > f(s_{K+2}) > \cdots > f(s_{K+L}) \\ >&f(s_1) > f(s_2) > \cdots > f(s_K) > f(s_{K+L+1}). \end{aligned} \quad (18)$$

Given a $(1+1)$ elitist EA to maximise a function f, f is called a *two-paths-I function* to the $(1+1)$ EA if there exist two paths to the optimum: $\text{Path}_1(s_{K+L+1}, s_K, \cdots, s_1, s_0)$ and $\text{Path}_2(s_{K+L+1}, s_{K+L}, \cdots, s_{K+1}, s_0)$ such that

- for $k = 1, \cdots, K$ and $k = K+2, \cdots, K+L$, mutation probabilities $P_{\text{m}}(s_k, s_{k-1}) = 1$;
- for $k = K+1$, mutation probability $P_{\text{m}}(s_k, s_0) = 1$;
- for $k = K+L+1$, mutation probabilities $P(s_k, s_K) = p$ and $P_{\text{m}}(s_k, s_{K+L}) = 1-p$ where $0 < p < 1$;
- for any other i, j, $P_{\text{m}}(s_i, s_j) = 0$.

Fig. 2 visualises a two-paths-I time-fitness landscape.

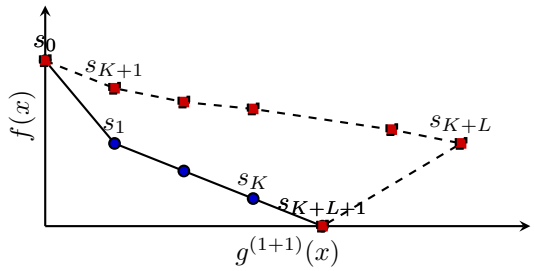

Fig. 2. A two-paths-I time-fitness landscape. The x axis represents the expected hitting time of the $(1+1)$ EA. The y axis is the fitness function.

Consider a $(1+\lambda)$ EA (where $\lambda \geq 2$) for maximizing a two-paths-I function function.

- **Mutation.** Mutation probabilities are identical to those in the $(1+1)$ EA. Generate λ children.
- **Elitist Selection.** Select the best individual from $\Phi_t \cup \Psi_t$.

The theorem below shows that using a population will increase the expected hitting time if the EA starts at s_{K+L+1}.

Proposition 1: Given the $(1+1)$ elitist EA and a $(1+\lambda)$ EA (where $\lambda \geq 2$) for maximizing a two-paths-I function, let $\Phi_0^{(1+1)} = \Phi_0^{(1+\lambda)} = s_{K+L+1}$, then $PS(1+\lambda \mid s_{K+L+1}) < 1$. The scalability threshold is 2.

Proof: For the $(1+\lambda)$ EA, let its distance function $d^{(1+\lambda)}(x) = g^{(1+1)}(x)$.

Consider the pointwise drift at s_{K+L+1}. There are two potential events at S_{K+1+1}.

1) The $(1+\lambda)$ EA moves from s_{K+L+1} to s_K. This event happens if and only if all children are s_K. The probability for the event happening is p^λ.
2) The $(1+\lambda)$ EA moves from s_{K+L+1} to s_{K+L}. This event happens if and only if at least one child is s_{K+L}. The probability for the event happening is $1-p^\lambda$.

We calculate the pointwise drift at s_{K+L+1} as follows:

$$\begin{aligned} &\Delta^{(1+\lambda)}(s_{K+L+1}) \\ =&(1-p^\lambda)(g^{(1+1)}(s_{K+L+1}) - g^{(1+1)}(s_{K+L})) \\ &+ p^\lambda(g^{(1+1)}(s_{K+L+1}) - g^{(1+1)}(s_K)) \\ =&(1-p^\lambda)[1 + pK + (1-p)L - L] \\ &+ p^\lambda[1 + pK + (1-p)L - K] \\ =&1 + (p - p^\lambda)(K - L) < 1. \\ &\text{(since } L > K, p \in (0,1) \text{ and } \lambda \geq 2) \end{aligned} \quad (19)$$

For any other non-optimal $s \in \{s_1, \cdots, s_{K+L}\}$, we calculate the pointwise drift $\Delta^{(1+\lambda)}(S_k) = 1$ from $P(s_k, s_{k-1}) = 1$ (except $k = K+1$) and $P(s_{K+1,0}) = 1$.

Since $\Phi_0^{(1+\lambda)} = s_{K+L+1}$, average drift $\bar{\Delta}_0^{(1+\lambda)} < 1$. For any $t \geq 1$, $\Phi_t^{(1+\lambda)}$ has left the point s_{K+L+1}, then average drift $\bar{\Delta}_t^{(1+\lambda)} = 1$. According to Theorem 8, we get that $PS(1+\lambda \mid s_{K+L+1}) < 1$. ∎

It is easy to understand the proposition. There are two paths towards the optimum: short and long. If using a population, the long path is more likely to be chosen than the short path. Then the expected hitting time is increased.

Example 1: Consider an instance of two-paths-I functions. Let $x \in \{0,1\}^n$ and $|x|$ denote its number of 1-valued bits.

$$f(x) = \begin{cases} n, & \text{if } |x| = 0 \text{ or } |x| = n, \\ -|x|, & \text{if } |x| \leq \theta \text{ where } \theta = 2, \\ |x|, & \text{otherwise.} \end{cases} \quad (20)$$

There are two optima: $|x| = 0, n$. Let x_0 be a string such that $|x_0| = 2$. It takes the minimum value of the fitness. A $(1+\lambda)$ EA uses adaptive mutation and elitist selection for solving the above problem.

- **Adaptive mutation.**
  1) When $t = 0$, flip one of 1-valued bits with the probability 0.5, otherwise flip one of 0-valued bits. In this way, generate λ children as Ψ_0.
  2) When $t \geq 1$, if $f(\Phi_t) > f(\Phi_{t-1})$ and Φ_t is generated by flipping a 0-valued bit (or 1-valued) in Φ_{t-1}, flip a 0-valued bit (or 1-value bit). Then generate λ children as Ψ_t.
  3) When $t \geq 1$, if $f(\Phi_t) = f(\Phi_{t-1})$, either flip one of 1-valued bits with probability 0.5 or flip one of 0-valued bits. Then generate λ children as Ψ_t.
- **Elitist Selection.** Select the best individual from $\Phi_t \cup \Psi_t$ as Φ_{t+1}.

f is a unimodal function under Hamming distance but a two-paths-I function (multi-modal) to the $(1+1)$ EA on the time-fitness landscape. When the $(1+1)$ EA starts at $|x| = 2$, there

are two paths towards the optima $|x| = 0, n$. The short path is $|x| = 2 \to 1 \to 0$. The long path is $|x| = 2 \to 3 \to \cdots \to n$. Table I shows that using a population increases the expected hitting time.

TABLE I
EXPERIMENTAL RESULTS FOR EXAMPLE 1 AVERAGED OVER 1000 RUNS. $n = 1000$. THE EA STARTS AT x_0 WITH $|x_0| = 2$.

population size	1	3	5	7	9
hitting time	491	872	961	990	997
running time	491	2616	4805	6930	8973

B. Case study 2: unimodal functions

This subsection presents another case study to show that using a population can not reduce the expected hitting time. The case study discusses a $(\mu + 1)$ elitist EA (where $\mu \geq 1$) with global mutation for maximizing any unimodal function.

- **Global Mutation.** Choose one individual from Φ_t at random and generate a child by mutation. Mutation probability $P_{\mathrm{m}}(s_i, s_j) > 0$ for any i and j.
- **Elitist Selection.** If the child's fitness is better than that of one or more parents, then the child will replace one of these parents at random.

The proposition below asserts that using a population increases the expected hitting time of the $(\mu + 1)$ EA.

Proposition 2: Given the $(1 + 1)$ EA and a $(\mu + 1)$ EA (where $\mu \geq 2$) for maximizing a unimodal function, for any $x \in \{s_2, \cdots, s_K\}$, let $\Phi_0^{(1+1)} = x$ and $\Phi_0^{(\mu+1)} = (x, \cdots, x)$, then $PS(\mu + 1 \mid x) < 1$. The scalability threshold is 2.

Proof: For the $(1 + 1)$ EA, choose $g^{(1+1)}(x)$ to be its distance function. According to Theorem 1, the pointwise drift satisfies $\Delta^{(1+1)}(s_k) = 1$ for any $s_k \in \{s_1, \cdots, s_K\}$. Since selection is elitist and the function is unimodal, we have

$$\Delta^{(1+1)}(s_k) = \textstyle\sum_{l=0}^{k-1} P_{\mathrm{m}}(s_k, s_l)\big(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)\big) = 1. \quad (21)$$

For the $(\mu+1)$ EA, define its distance function $d^{(\mu+1)}(X) = \min\{g^{(1+1)}(x) : x \in X\}$.

Let X be a parent population: $X = (s_{1(X)}, \cdots, s_{\mu(X)}) \in \mathcal{P}_{\mathrm{non}}$ in which the best parent is s_k. Let their child be s_l. Since the $(\mu+1)$ EA adopts elitist selection and the fitness function is unimodal, the pointwise drift satisfies

$$\Delta^{(\mu+1)}(X) = \textstyle\sum_{l=0}^{k-1} P_{\mathrm{m}}(X, s_l)\big(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)\big). \quad (22)$$

The probability of mutating a parent $s_{i(X)}$ to the child s_l (where $l < k$) is $P_{\mathrm{m}}(s_{i(X)}, s_l)$. Since each parent is chosen for mutation at random, we have

$$P_{\mathrm{m}}(X, s_l) = \frac{1}{\mu} \sum_{i=1}^{\mu} P_{\mathrm{m}}(s_{i(X)}, s_l).$$

Then we get

$$\begin{aligned} &P_{\mathrm{m}}(X, s_l)\left(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)\right) \\ =&\tfrac{1}{\mu}\textstyle\sum_{i=1}^{\mu} P_{\mathrm{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)). \end{aligned}$$

Since $l < k \leq i(X)$, according to the definition of unimodal functions, we have

$$g^{(1+1)}(s_l) < g^{(1+1)}(s_k) \leq g^{(1+1)}(s_{i(X)}),$$

and then

$$\begin{aligned} &P_{\mathrm{m}}(X, s_l)\left(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)\right) \\ \leq&\tfrac{1}{\mu}\textstyle\sum_{i=1}^{\mu} P_{\mathrm{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_{i(X)}) - g^{(1+1)}(s_l)) \end{aligned}$$

The pointwise drift satisfies

$$\begin{aligned} &\Delta^{(\mu+1)}(X) \\ =&\tfrac{1}{\mu}\textstyle\sum_{i=1}^{\mu}\sum_{l=0}^{k-1} P_{\mathrm{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)) \\ \leq&\tfrac{1}{\mu}\textstyle\sum_{i=1}^{\mu}\sum_{l=0}^{k-1} P_{\mathrm{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_{i(X)}) - g^{(1+1)}(s_l)). \end{aligned} \quad (23)$$

1) **Case 1:** for all parents in X, $s_{i(X)} = s_k$. According to Equality (21), we have

$$\textstyle\sum_{l=0}^{k-1} P_{\mathrm{m}}(s_k, s_l)(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)) = 1.$$

Then the pointwise drift satisfies

$$\Delta^{(\mu+1)}(X) = \frac{1}{\mu}\sum_{i=1}^{\mu} 1 = 1. \quad (24)$$

2) **Case 2:** for at least one parent in X, $s_{i(X)} \neq s_k$. Since s_k is the best parent in X, the indexes satisfy $i(X) > k$. Thanks to global mutation, we have $P_{\mathrm{m}}(s_{i(X)}, s_l) > 0$ for any $i(X)$, and then

$$\begin{aligned} &\textstyle\sum_{l=0}^{k-1} P_{\mathrm{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_{i(X)}) - g^{(1+1)}(s_l)) \\ <&\textstyle\sum_{l=0}^{i(X)-1} P_{\mathrm{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_{i(X)}) - g^{(1+1)}(s_l)) \\ =&\Delta^{(1+1)}(s_{i(X)}) = 1. \end{aligned} \quad (25)$$

The last equality "$= 1$" comes from Equality (21). Then the pointwise drift satisfies

$$\Delta^{(\mu+1)}(X) < \frac{1}{\mu}\sum_{i=1}^{\mu} 1 = 1. \quad (26)$$

Since $\Phi_0 = (s_k, \cdots, s_k)$ where $k \geq 2$, the average drift $\bar{\Delta}_0^{(\mu+1)} = 1$. When $t = 1$, the probability of $\Phi_1 \in \mathcal{P}_{\mathrm{non}}$ including two different non-optimal points is always greater than 0 due to global mutation. Thus the average drift $\bar{\Delta}_1^{(\mu+1)} < 1$. When $t \geq 2$, the average drift $\bar{\Delta}_t^{(\mu+1)} \leq 1$. According to Theorem 8, we get $PS(\mu + 1 \mid s_k) < 1$. ■

Here is an explanation of this proposition. For unimodal functions, the higher the fitness of a point is, the closer to the optimal set the point is (Fig. 1). Given a parent population in the $(\mu+1)$ EA, a good strategy is to mutate the best parent in the population. Unfortunately, a population may include some parent which is worse than the best. If a worse parent is chosen to mutate, it will increase he expected hitting time.

Example 2: Consider the TwoMax function [32]. Let $x \in \{0, 1\}^n$ and $|x|$ denote the number of 1-valued bits.

$$f(x) = \max\{|x|, n - |x|\}. \quad (27)$$

There are two optima: $|x| = 0, n$. A $(\mu+1)$ elitist EA is used for solving the maximization problem.

- **Bitwise Mutation.** Choose one individual from μ parents at random. Flip each bit with a probability $1/n$.

- **Elitist Selection.** If the child's fitness is better than that of one or more parents, then the child will replace one of these parents at random.

f is unimodal to the $(1+1)$ EA [32]. Table II shows that using a population increases the expected hitting time.

TABLE II
EXPERIMENTAL RESULTS FOR EXAMPLE 2 AVERAGED OVER 1000 RUNS. $n = 200$. THE EA STARTS AT x_0 WITH $|x_0| = 100$.

population size	1	3	5	7	9
hitting time	2523	2650	3014	3477	3855
running time	2523	2650	3014	3477	3855

V. SCENARIO 2: USING A POPULATION CAN REDUCE HITTING TIME BUT NOT RUNNING TIME

A. Case study 3: unimodal functions

Let's reinvestigate the intuitive belief that using a population can reduce the expected hitting time of elitist EAs for maximizing unimodal functions. Although this belief is not true for the $(\mu+1)$ EA, it is still true for the $(\lambda+\lambda)$ EA with global mutation and elitist selection.

Consider a $(\lambda+\lambda)$ EA ($\lambda \geq 1$) using elitist selection and global mutation for maximising a unimodal function.

- **Global Mutation.** Mutation probability $P_{\mathrm{m}}(s_i, s_j) > 0$ for any i, j. Each individual in Φ_t generates a child.
- **Elitist Selection.** Probabilistically select λ individuals from $\Phi_t \cup \Psi_t$, while the best individual is always selected.

First we prove an inequality which will be used later.

Lemma 1: Given $a_i > 0, b_i > 0, c_i > 0$ where $i = 0, 1, \cdots, k$ such that $\sum_{i=0}^{j} a_i > \sum_{i=0}^{j} b_i$, $j = 0, \cdots, k$ and $c_0 > c_1 > \cdots > c_k$, it holds $\sum_{i=0}^{k} a_i c_i > \sum_{i=0}^{k} b_i c_i$.

Proof: From the conditions $a_0 + \cdots + a_j > b_0 + \cdots + b_j$ and $c_0 > c_1 > \cdots > c_k$, we have

$$\begin{aligned} & \textstyle\sum_{i=0}^{k}(a_i - b_i)c_i \\ = \;& (a_0 - b_0)c_0 + \textstyle\sum_{i=1}^{k}(a_i - b_i)c_i \\ > \;& (a_0 - b_0)c_1 + \textstyle\sum_{i=1}^{k}(a_i - b_i)c_i \\ = \;& (a_0 - b_0 + a_1 - b_0)c_1 + \textstyle\sum_{i=2}^{k}(a_i - b_i)c_i \\ > \;& (a_0 - b_0 + a_1 - b_1)c_2 + \textstyle\sum_{i=2}^{k}(a_i - b_i)c_i \\ = \;& (a_0 - b_0 + a_1 - b_0 + a_2 - b_2)c_2 + \textstyle\sum_{i=3}^{k}(a_i - b_i)c_i. \end{aligned}$$

By induction, we can prove that

$$\begin{aligned} & \textstyle\sum_{i=0}^{k}(a_i - b_i)c_i \\ > \;& (a_0 - b_0 + a_1 - b_1 + \cdots + a_k - b_k)c_k > 0. \end{aligned}$$

This gives the desired result. ∎

Proposition 3: Given the $(1+1)$ elitist EA and a $(\lambda+\lambda)$ EA (where $\lambda \geq 2$) using global mutation and elitist selection for maximizing any unimodal function, let $\Phi_0^{(1+1)} = x$ and $\Phi_0^{(\lambda+\lambda)} = (x, \cdots, x)$ where $x \in \mathcal{P}_{\mathrm{non}}$, then $PS(\lambda+\lambda \mid x) > 1$.

Proof: For the $(1+1)$ EA, choose $g^{(1+1)}(x)$ to be its distance function. According to Theorem 1, the pointwise drift satisfies $\Delta^{(1+1)}(s_k) = 1$ for any $s_k \in \mathcal{P}_{\mathrm{non}}$. Since selection is elitist and the function is unimodal, we have

$$\Delta^{(1+1)}(x) = \textstyle\sum_{l=0}^{k-1} P_{\mathrm{m}}(s_k, s_l)(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)) = 1. \quad (28)$$

For a $(\lambda+\lambda)$ EA, define its distance function $d^{(\lambda+\lambda)}(X) = \min\{g^{(1+1)}(x) : x \in X\}$.

Let $\mathcal{P}_k$ denote the set of populations whose best individual is s_k (where $k = 1 \cdots, K$). Provided that the parent population is $X \in \mathcal{P}_k$ (where $k > 0$), the children population is Y and the next generation population is $Z \in \mathcal{P}_l$.

Since the $(\mu+\lambda)$ EA adopts elitist selection and the fitness function is unimodal, the pointwise drift satisfies

$$\Delta^{(\mu+\lambda)}(X) = \textstyle\sum_{l=0}^{k-1} \sum_{Z \in \mathcal{P}_l} P(X,Z)\big(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)\big). \quad (29)$$

Given any $m < k$, the children population Y and next generation population enter in the union $\mathcal{P}_0 \cup \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_m$ if and only if one or more parents in X is mutated into an child in the set $\{s_0, \cdots, s_m\}$. Thanks to global mutation and population size $\lambda \geq 2$, this probability is strictly larger than that of only one parent s_k being mutated into the set $\{s_0, \cdots, s_m\}$:

$$\sum_{l=0}^{m} \sum_{Z \in \mathcal{P}_l} P(X, Z) > \sum_{l=0}^{m} P(s_k, s_l).$$

Since the fitness function is unimodal, we have

$$\begin{aligned} & g^{(1+1)}(s_k) - g^{(1+1)}(s_0) > g^{(1+1)}(s_k) - g^{(1+1)}(s_1) \\ & > \cdots > g^{(1+1)}(s_k) - g^{(1+1)}(s_{k-1}). \end{aligned}$$

Using Lemma 1 (let $a_l = \sum_{Z \in \mathcal{P}_l} P(X, Z)$, $b_l = P(s_k, s_l)$ and $c_l = g^{(1+1)}(s_k) - g^{(1+1)}(s_l)$), we get

$$\begin{aligned} \Delta^{(\lambda+\lambda)}(X) &= \sum_{l=0}^{k-1} \sum_{Z \in \mathcal{P}_l} P(X, Z)(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)) \\ &> \sum_{l=0}^{k-1} P(s_k, s_l)(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)) \\ &= \Delta^{(1+1)}(s_k) = 1. \end{aligned}$$

Then we have $\Delta^{(\lambda+\lambda)}(X) > 1$. Since $\Phi_0 \in \mathcal{P}_{\mathrm{non}}$, we have for any $t \geq 0$, the average drift $\bar{\Delta}_t^{(\lambda+\lambda)} > 1$. Applying Theorem 9, we get $PS(\lambda+\lambda) > 1$. ∎

Here is an explanation of the proposition. For unimodal functions, the higher the fitness of a point is, the closer to the optimal set the point is. The probability of the $(\lambda+\lambda)$ EA (where $\lambda \geq 2$) to generate a better individual is strictly larger than that of the $(1+1)$ EA. Thus the expected hitting time is shortened.

B. Case study 4: unimodal functions

It is an intuitive belief that using a population can not reduce the expected running time of elitist EAs for maximizing unimodal functions. The proposition below asserts this is true for unimodal functions on the time-fitness landscape.

Consider a $(\mu+\lambda)$ elitist EA.

- **Mutation.** Select λ individuals in Φ_t and mutate them. Then generate a children population consisting of λ individuals;

- **Elitist Selection.** First select one individual with the highest fitness in $\Phi_t \cup \Psi_t$; and then probabilistically select $\mu - 1$ individuals from $\Phi_t \cup \Psi_t$.

Proposition 4: Given the $(1+1)$ elitist EA and a $(\mu+\lambda)$ EA (where $\mu \geq 2$ or $\lambda \geq 2$) for maximizing a unimodal function, let $\Phi_0^{(1+1)} = x$ and $\Phi_0^{(\mu+\lambda)} = (x, \cdots, x)$ where $x \in \mathcal{P}_{\text{non}}$, then $PS(\mu+\lambda \mid x) \leq \lambda$ and if $\lambda \geq 2$, $PS(\mu+\lambda \mid x) < \lambda$.

Proof: It is sufficient to consider the case of $\lambda > 1$. The analysis of the $(\mu+1)$ EA is almost the same as that of Proposition 2, except two places: (1) without the global mutation condition, Inequality (25) is changed from $<$ to $\leq$; (2) the conclusion is changed from $PS(\mu+1) < 1$ to $PS(\mu+1) \leq 1$.

For the $(1+1)$ EA, choose $g^{(1+1)}(x)$ to be its distance function. According to Theorem 1, for any $s_k \in \{s_1, \cdots, s_K\}$, the pointwise drift satisfies $\Delta^{(1+1)}(s_k) = 1$. Since selection is elitist and the function is unimodal, we have

$$\Delta^{(1+1)}(s_k) = \textstyle\sum_{l=0}^{k-1} P_{\text{m}}(s_k, s_l)(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)) = 1. \quad (30)$$

For the $(\mu+\lambda)$ EA where $\lambda \geq 2$, let its distance function $d^{(\mu+\lambda)}(X) = \min\{g^{(1+1)}(x) : x \in X\}$.

Let $\mathcal{P}_k$ denote the set of populations whose best individual is s_k (where $k = 0, 1, \cdots, K$). Provided that the parent population is $X \in \mathcal{P}_k$ (where $k > 0$), the children population is Y and the next generation population is $Z \in \mathcal{P}_l$.

Since the $(\mu+\lambda)$ EA adopts elitist selection and the fitness function is unimodal, the pointwise drift satisfies

$$\begin{aligned}&\Delta^{(\mu+\lambda)}(X)\\ &= \textstyle\sum_{l=0}^{k-1} \sum_{Z \in \mathcal{P}_l} P(X,Z)\big(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)\big)\\ &= \textstyle\sum_{l=0}^{k-1} \sum_{Y \in \mathcal{P}_l} P_{\text{m}}(X,Y)\big(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)\big). \end{aligned} \quad (31)$$

Denote the children population Y by $(s_{1(Y)}, \cdots, s_{\lambda(Y)})$. Let $(s_{1(X)}, \cdots, s_{\lambda(X)})$ be the parents from which Y are mutated. Children $Y \in \mathcal{P}_l$ (where $l < k$) only if one or more parents is muted into s_l. The probability of mutating $s_{i(X)}$ to s_l is $P_{\text{m}}(s_{i(X)}, s_l)$. Since each parent is mutated independently, the probability of one or more parents is muted into s_l is not more than the sum of each parents is mutated into s_l. Then we have

$$\begin{aligned}&\textstyle\sum_{Y \in \mathcal{P}_l} P_{\text{m}}(X,Y)\left(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)\right)\\ &\leq \textstyle\sum_{i=1}^{\lambda} P_{\text{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)). \end{aligned} \quad (32)$$

The above inequality is strict if $X = (s_k, \cdots, s_k)$.

Since $l < k \leq i(X)$, we have

$$g^{(1+1)}(s_l) < g^{(1+1)}(s_k) \leq g^{(1+1)}(s_{i(X)}),$$

and then

$$\begin{aligned}&\textstyle\sum_{Y \in \mathcal{P}_l} P_{\text{m}}(X,Y)\left(g^{(1+1)}(s_k) - g^{(1+1)}(s_l)\right)\\ &\leq \textstyle\sum_{i=1}^{\lambda} P_{\text{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_{i(X)}) - g^{(1+1)}(s_l)).\end{aligned}$$

Inserting the above inequality into Equality (31), we get

$$\begin{aligned}&\Delta^{(\mu+\lambda)}(X)\\ &\leq \textstyle\sum_{l=0}^{k-1} \sum_{i=1}^{\lambda} P_{\text{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_{i(X)}) - g^{(1+1)}(s_l))\\ &= \textstyle\sum_{i=1}^{\lambda} \sum_{l=0}^{k-1} P_{\text{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_{i(X)}) - g^{(1+1)}(s_l)). \end{aligned} \quad (33)$$

Since s_k is the best parent in X, we have for $i = 1, \cdots, \lambda$, the indexes satisfy $i(X) \geq k$. Then

$$\begin{aligned}&\textstyle\sum_{l=0}^{k-1} P_{\text{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_{i(X)}) - g^{(1+1)}(s_l))\\ &\leq \textstyle\sum_{l=0}^{i(X)-1} P_{\text{m}}(s_{i(X)}, s_l)(g^{(1+1)}(s_{i(X)}) - g^{(1+1)}(s_l)) = 1.\end{aligned}$$

The drift satisfies

$$\Delta^{(\mu+\lambda)}(X) \leq \sum_{i=1}^{\lambda} 1 = \lambda.$$

The above inequality is strict if $X = (s_k, \cdots, s_k)$.

Since $\Phi_0 = (s_k, \cdots, s_k)$ for some $k \geq 1$, the average drift $\bar{\Delta}_0^{(\mu+\lambda)} < 1$. When $t \geq 1$, the average drift $\bar{\Delta}_t^{(\mu+\lambda)} \leq 1$. Applying Theorem 8, we obtain $PS(\mu+\lambda) < \lambda$. ■

The explanation of the proposition is simple. For unimodal functions, the higher the fitness of a point is, the closer to the optimal set the point is. The probability of the $(\mu+\lambda)$ EA to generate a better individual is not more than λ times that of the $(1+1)$ EA. Thus the expected hitting time cannot be shortened by $1/\lambda$.

Example 3: Consider the TwoMax function [32] where $x \in \{0,1\}^n$.

$$f(x) = \max\{|x|, n - |x|\}. \quad (34)$$

A $(1+\lambda)$ EA (where $\lambda \geq 1$) with elitist selection and bitwise mutation is used for maximizing the function.

- **Bitwise Mutation.** Flip each bit with a probability $1/n$. Then generates λ children.
- **Elitist Selection.** Select the best individual from $\Phi_t \cup \Psi_t$.

Table III shows that using a population reduces the expected hitting time, but increases the expected running time.

TABLE III
EXPERIMENTAL RESULTS FOR EXAMPLE 3 AVERAGED OVER 1000 RUNS. $n = 200$. THE EA STARTS AT x_0 WITH $|x_0| = 100$.

population size	1	3	5	7	9
hitting time	2536	864	529	395	315
running time	2536	2592	2645	2765	2835

Let's apply Proposition 4 to a special instance: the $(1+\lambda)$ EA (using bitwise mutation and elitist selection) for maximizing the OneMax function. According to Proposition 4, using a population will increase the running time. The population threshold is 2. This conclusion is more accurate that that in [8]. The result in [8] asserts that the expected running time of the $(1+\lambda)$ EA is in the same order of that of the $(1+1)$ EA by a constant factor when λ is smaller than the cut-off point. The constant could be $= 1$, > 1 or < 1. But the two results are not contrary. Our result indicates that the constant factor is strictly less than 1 when $\lambda \geq 2$.

VI. SCENARIO 3: USING A POPULATION CAN REDUCE RUNNING TIME

A. Case study 5: two-paths-II functions

In the previous section, it has been proven that using a population cannot reduce the expected running time of an

EA for maximizing any unimodal function on the time-fitness landscape. But this intuitive belief is not true in terms of the distance-based fitness landscape. It is demonstrated by the following case study of two-paths-II functions.

Definition 7: Let $\mathcal{S} = \{s_0, s_1, \cdots, s_{K+L}, s_{K+L+1}\}$ where $L < K$ and f a fitness function such that

$$\begin{aligned} &f(s_0) > f(s_{K+1}) > f(s_{K+2}) > \cdots > f(s_{K+L}) \\ >&f(s_1) > f(s_2) > \cdots > f(s_K) > f(s_{K+L+1}). \end{aligned} \tag{35}$$

Given a $(1+1)$ elitist EA to maximise f, f is called a *two-paths-II function* to the $(1+1)$ EA if there exist two paths to the optimum: $\text{Path}_1(s_{K+L+1}, s_K, \cdots, s_1, s_0)$ and $\text{Path}_2(s_{K+L+1}, s_{K+L}, \cdots, s_{K+1}, s_0)$ such that

- for $k = 1, \cdots, K$ and $k = K+2, \cdots, K+L$, mutation probabilities $P_{\text{m}}(s_k, s_{k-1}) = 1$;
- for $k = K+1$, mutation probability $P_{\text{m}}(s_k, s_0) = 1$;
- for $k = K+L+1$, mutation probabilities $P(s_k, s_K) = p$ and $P_{\text{m}}(s_k, s_{K+L}) = 1-p$ where $0 < p < 1$;
- for any other i, j, $P_{\text{m}}(s_i, s_j) = 0$.

Fig. 3 visualises the a two-paths-II time-fitness landscape.

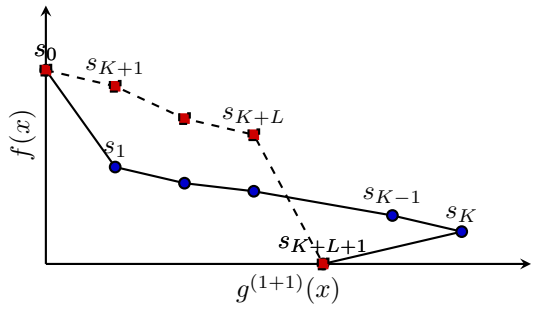

Fig. 3. A two-paths-II time-fitness landscape. The x axis represents the expected hitting time of the $(1+1)$ EA. The y axis is the fitness function.

Consider a $(1+\lambda)$ EA where $\lambda \geq 2$.

- **Mutation.** Mutation probabilities are identical to those in the $(1+1)$ EA in the above definition.
- **Elitist Selection.** Select the best individual from $\Phi_t \cup \Psi_t$.

Under certain condition, using a population may reduce the expected hitting time.

Proposition 5: Given the $(1+1)$ EA and a $(1+\lambda)$ EA (where $\lambda \geq 2$) for maximizing a two-paths-II function, let $\Phi_0^{(1+1)} = s_{K+L+1}$ and $\Phi_0^{(1+\lambda)} = s_{K+L+1}$, if the population size satisfies $\lambda < \lambda^*$ where λ^* is given by

$$\lambda^* = 1 + (p - \lambda p^\lambda)\frac{K-L}{L+1}, \tag{36}$$

then $PS(1+\lambda \mid s_{K+L+1}) > \lambda$. The scalability threshold is not less than λ^*.

Proof: For the $(1+\lambda)$ EA, define its distance function as follows:

$$d^{(1+\lambda)}(x) = \begin{cases} 0, & \text{if } x = s_0, \\ g^{(1+1)}(s_{K+L+1}), & \text{if } x = s_{K+L+1}, \\ \lambda g^{(1+1)}(x), & \text{otherwise.} \end{cases} \tag{37}$$

Consider the pointwise drift at s_{K+L+1}. There are two potential events.

1) The $(1+\lambda)$ EA moves from s_{K+L+1} to s_K. This event happens if and only if all mutated children are s_K. The probability for the event happening is p^λ.
2) The $(1+\lambda)$ EA moves from s_{K+L+1} to s_{K+L}. This event happens if and only if at least one mutated child is s_{K+L}. The probability for the event happening is $1-p^\lambda$.

Calculate the pointwise drift at s_{K+L+1} as follows:

$$\begin{aligned} &\Delta^{(1+\lambda)}(s_{K+L+1}) \\ =&(1-p^\lambda)(g^{(1+1)}(s_{K+L+1}) - \lambda g^{(1+1)}(s_{K+L})) \\ &+ p^\lambda(g^{(1+1)}(s_{K+L+1}) - \lambda g^{(1+1)}(s_K)) \\ =&(1-p^\lambda)[1 + pK + (1-p)L - \lambda L] \\ &+ p^\lambda[1 + pK + (1-p)L - \lambda K] \\ =&1 + pK + (1-p)L - \lambda L + p^\lambda \lambda L - p^\lambda \lambda K \\ =&1 + (p - p^\lambda\lambda)(K-L) + L(1-\lambda) \\ >&\lambda. \quad (\text{use } \lambda < \lambda^* \text{ and } (36)) \end{aligned} \tag{38}$$

Calculate the pointwise drift at $s \in \{s_1, \cdots, s_{K+L}\}$,

$$\begin{aligned} \Delta^{(1+\lambda)}(s) &= \sum_{s' \in \mathcal{S}} [\lambda g^{(1+1)}(s) - \lambda g^{(1+1)}(s')] \\ &= \Delta^{(1+1)}(s) = \lambda. \end{aligned}$$

Since $\Phi_0 = s_{K+L+1}$, the average drift $\bar{\Delta}_0^{(1+\lambda)} > \lambda$. For any $t \geq 1$, $\Phi_t^{(1+\lambda)}$ has left the point s_{K+L+1}, then average drift $\bar{\Delta}_t^{(1+\lambda)} = \lambda$. According to Theorem 9, we get that $PS(1+\lambda \mid s_{K+L+1}) > \lambda$. ■

It is easy to understand the proposition. There are two paths to the optimum: short and long. Using a lager population, the short path is more likely to be chosen than the long path. Thus the expected hitting time is reduced.

Example 4: Consider an instance of two-paths-II functions. Let $x \in \{0,1\}^n$ and $|x|$ denote its number of 1-valued bits.

$$f(x) = \begin{cases} n, & \text{if } |x| = 0 \text{ or } |x| = n, \\ -|x|, & \text{if } |x| \leq \theta \text{ where } \theta = n-2, \\ |x|, & \text{otherwise.} \end{cases} \tag{39}$$

The $(1+\lambda)$ EA is the same as that in Example 1 in Section IV-A. Table IV shows that using a population reduces the expected running time.

TABLE IV
EXPERIMENTAL RESULTS FOR EXAMPLE 4 AVERAGED OVER 1000 RUNS. $n = 1000$. THE EA STARTS AT x_0 WITH $|x_0| = 998$.

population size	1	5	9	13	17
hitting time	507	28	2	2	2
running time	507	140	18	26	34

f is a unimodal function under Hamming distance but a two-path-II function (multi-modal) to the $(1+1)$ EA on the time-fitness landscape. When the $(1+1)$ EA starts at $|x| = n-2$, there are two paths towards the optima $|x| = 0, n$. The long path is $|x| = n-2 \to n-3 \to \cdots \to 0$. The short path is $|x| = n-2 \to n-1 \to n$. This example shows that using a population shortens the expected running time of an EA on a unimodal function in terms of the distance-based fitness

landscape. It doesn't contradict Proposition 4, which holds for any unimodal functions on the time-fitness functions.

B. Case study 6: deceptive-like functions

It is an intuitive belief that using a population may shorten the runtime of EAs on deceptive functions. This was proven for an elitist EA on a deceptive function under Hamming distance [7]. In this case study, the conclusion is generalised to deceptive-like functions in any finite set. Deceptive functions and deceptive-like functions are defined on the time-fitness landscape [32] .

Definition 8: Let $\mathcal{S} = \{s_0, s_1, \cdots, s_K\}$ and f a fitness function such that $f(s_0) > f(s_K) > \cdots > f(s_1)$. Given a $(1+1)$ elitist EA to maximise f, f is called *deceptive* to the $(1+1)$ EA if $g^{(1+1)}(s_K) > \cdots > g^{(1+1)}(s_1)$.

According to [32], deceptive functions to a $(1+1)$ elitist EA are the hardest among all fitness functions with the same optimum at s_0. Furthermore, according to [32], for any fitness function $f(x)$, we can construct a $(1+1)$ elitist EA to which f is the hardest (deceptive).

Definition 9: Let $\mathcal{S} = \{s_0, s_1, \cdots, s_K\}$ and f a fitness function such that

$$f(s_0) > f(s_K) > \max\{f(s_1), \cdots, f(s_{K-1})\}. \tag{40}$$

Given a $(1+1)$ elitist EA to maximise f, f is called *deceptive-like* to the $(1+1)$ EA if $g^{(1+1)}(s_K) > g^{(1+1)}(s_k)$ for any $k < K$.

A deceptive-like time-fitness landscape is visualised in Fig. 4. Deceptive functions belong to deceptive-like functions.

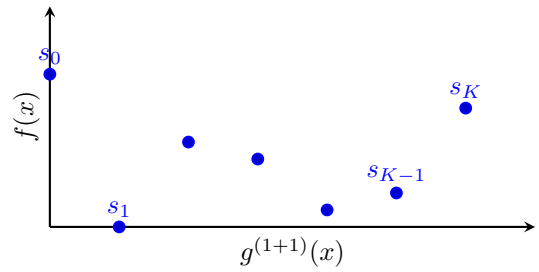

Fig. 4. A deceptive-like time-fitness landscape. The x axis is the expected hitting time of the $(1+1)$ EA. The y axis is the fitness function.

Given a fitness function $f(x)$, consider an elitist $(1+1)$ uses global mutation for maximising $f(x)$.

- **Global Mutation.** Mutation probability $P_m(s_i, s_j) > 0$ for any i, j and mutation probability $P_m(s_i, s_0) > P_{\mathrm{m}}(s_K, s_0)$ for any $i < K$.
- **Elitist Selection.** select the best from $\Phi_t \cup \Psi_t$.

The fitness function f is deceptive-like to the $(1+1)$ EA because $g^{(1+1)}(s_K) > g^{(1+1)}(s_i)$ for any $i < K$. This can be proven by pointwise drift analysis. Let the distance function to be $d^{(1+1)}(s) = g^{(1+1)}(s_K)$ for $s \in \{s_1, \cdots, s_K\}$ and $d^{(1+1)}(s) = 0$ for $s = s_0$. The pointwise drift satisfies $\Delta^{(1+1)}(s) = 1$ for $s = s_K$ and $\Delta^{(1+1)}(s_i) > 1$ for $s_i \in \{s_1, \cdots, s_{K-1}\}$ due to $P_m(s_i, s_0) > P_{\mathrm{m}}(s_K, s_0)$. Hence $g^{(1+1)}(s_K) > g^{(1+1)}(s_i)$ (deceptive-like).

Consider a $(\lambda+\lambda)$ EA (where $\lambda \geq 2$) using global mutation and elitist selection with fitness diversity for maximising $f(x)$.

- **Global Mutation.** The same as that in the $(1+1)$ EA.
- **Elitist Selection with Fitness Diversity.**
  1) If all individuals in $\Phi_t \cup \Psi_t$ have the same fitness, then choose λ individuals at random;
  2) Otherwise, first select one individual with the highest fitness from $\Phi_t \cup \Psi_t$ and then select one individual with a lower higher fitness. Thereafter select $\lambda - 2$ individuals from $\Phi_t \cup \Psi_t$ using any selection scheme.

We will show that the running time of the $(\lambda+\lambda)$ EA (where $\lambda \geq 2$) is shorter than that of the $(1+1)$ EA. Before the theorem and its proof, the following notation is introduced.

$$\mathcal{S}_{\underline{K}} \stackrel{\text{def}}{=} \{x \in \mathcal{S}; f(x) < f(s_K)\}, \tag{41}$$

$$\mathcal{P}_{\underline{K}} \stackrel{\text{def}}{=} \{X \in \mathcal{P}; f(x) < f(s_K) \text{ for any } x \in X\}, \tag{42}$$

$$P_{\mathrm{m}}(x, \mathcal{S}_{\underline{K}}) \stackrel{\text{def}}{=} \sum_{y \in \mathcal{S}_{\underline{K}}} P_{\mathrm{m}}(x, y), \tag{43}$$

$$P_{\mathrm{m}}(X, \mathcal{P}_{\underline{K}}) \stackrel{\text{def}}{=} \sum_{x \in X} P_{\mathrm{m}}(x, \mathcal{S}_{\underline{K}}). \tag{44}$$

Proposition 6: Given the $(1+1)$ elitist EA and a $(\lambda+\lambda)$ EA (where $\lambda \geq 2$) for maximizing any deceptive-like function, if the population size satisfies $\lambda < \lambda^*$ where λ^* is given by

$$\lambda^* = \frac{P_{\mathrm{m}}(s_K, \mathcal{S}_{\underline{K}}) \min_{0<i<K} P_{\mathrm{m}}(s_i, s_0)}{(P_{\mathrm{m}}(s_K, \mathcal{S}_{\underline{K}}) + \min_{0<i<k} P_{\mathrm{m}}(s_i, s_0)) P_{\mathrm{m}}(s_K, s_0)}, \tag{45}$$

then $PS(\lambda+\lambda \mid s_K) > \lambda$. The scalability threshold is not less than λ^*.

Proof: For the $(\lambda+\lambda)$ EA, given a population X, let $f(X) = \max\{f(x); x \in X\}$ the maximal fitness of its individuals. Define the distance distance as follows:

$$d^{(\lambda+\lambda)}(X) = \begin{cases} d_0 = 0, & \text{if } f(X) = f(s_0), \\ d_K = g^{(1)}(s_K), & \text{if } f(X) = f(s_K), \\ d_{\underline{K}} = \frac{\lambda}{\min_{0<i<K} P_{\mathrm{m}}(s_i, s_0)}, & \text{if } f(X) < f(s_K). \end{cases} \tag{46}$$

We calculate the pointwise drift at $X = (s_K, \cdots, s_K)$:

$$\begin{aligned} &\Delta^{(\lambda+\lambda)}(s_K, \cdots, s_K) \\ =&P(X, \mathcal{P}_{\mathrm{opt}})(d_K - d_0) + P(X, \mathcal{P}_{\underline{K}})(d_K - d_{\underline{K}}) \\ >&P(X, \mathcal{P}_{\mathrm{opt}})(d_K - d_{\underline{K}}) + P(X, \mathcal{P}_{\underline{K}})(d_K - d_{\underline{K}}) \\ =&\left(P(X, \mathcal{P}_{\mathrm{opt}}) + P(X, \mathcal{P}_{\underline{K}})\right)(d_K - d_{\underline{K}}) \\ >&P_{\mathrm{m}}(s_K, \mathcal{S}_{\underline{K}})(d_K - d_{\underline{K}}) \\ =&P_{\mathrm{m}}(s_K, \mathcal{S}_{\underline{K}}) \left(\frac{1}{P_{\mathrm{m}}(s_K, s_0)} - \frac{\lambda}{\min_{0<i<K} P_{\mathrm{m}}(s_i, s_0)}\right) \\ >&\lambda. \qquad (\text{use } \lambda < \lambda^* \text{ and } (45)) \end{aligned} \tag{47}$$

Calculate the pointwise drift at any $X \in \mathcal{P}_{\mathrm{non}}$ in which at least one of its individuals is not s_K:

$$\begin{aligned} \Delta^{(\lambda+\lambda)}(X) =&P(X, \mathcal{P}_{\mathrm{opt}})(d_{\underline{K}} - d_0) \\ \geq&\frac{\lambda \max_{x \in X} P_{\mathrm{m}}(x, s_0)}{\min_{0<i<k} P_{\mathrm{m}}(s_i, s_0)} \geq \lambda. \end{aligned} \tag{48}$$

Since $\Phi_0^{(\lambda+\lambda)} = (s_K, \cdots, s_K)$, from (47) we have the average drift $\bar{\Delta}_0^{(\lambda+\lambda)} > \lambda$. For any $t \geq 1$, we know, $\bar{\Delta}_t^{(\lambda+\lambda)} \geq \lambda$. According to Theorem 9, we have $PS(\lambda+\lambda \mid s_K) > \lambda$. ■

The proposition can be explained as follows. In the $(1+1)$ algorithm, elitist selection cannot accept an worse solution, which is a bad strategy for deceptive-like functions. But in the population-based EA, selection with the fitness diversity can accept a worse solution. This helps the EA find the optimum more quickly.

Example 5: Consider an instance of deceptive functions. Let $x \in \{0,1\}^n$ and $|x|$ denote its number of 1-valued bits.

$$f(x) = \begin{cases} n, & \text{if } |x| = 0 \text{ or } |x| = n, \\ \min\{|x|, n-|x|\}, & \text{otherwise.} \end{cases} \quad (49)$$

There are two optima: $|x| = 0, n$. Consider a $(\lambda+\lambda)$ EA using elitist selection and bitwise mutation:

- **Bitwise Mutation.** Flip each bit with probability $1/n$. Each parent generates one child.
- **Elitist Selection + Random Selection.** Select one individual with the highest fitness from $\Phi_t \cup \Psi_t$ and then select $\lambda - 1$ individuals from $\Phi_t \cup \Psi_t$ at random.

According to [32], $f(x)$ is a deceptive function to the $(1+1)$ EA. Table V shows that using a population reduced the expected running time.

TABLE V
EXPERIMENTAL RESULTS FOR EXAMPLE 5 AVERAGED OVER 1000 RUNS. $n = 10$. THE EA STARTS AT x_0 WITH $|x_0| = 5$.

population size	1	5	9	13	17
hitting time	82774	1411	370	188	116
running time	82774	7055	3330	2444	1972

VII. DISCUSSION (CASE STUDY 7)

It must be pointed out that population scalability depends on the benchmark EA. Let's show this through a simple instance of deceptive functions. Let $x \in \{0,1\}^2$ be a binary string with length 2. The fitness function is given by

$$f(x) = \begin{cases} 3, & \text{if } |x| = 0, \\ |x|, & \text{if } |x| = 1, 2. \end{cases} \quad (50)$$

If the benchmark $(1+1)$ EA is changed from elitist selection to random selection, then using a population cannot shorten the running time any more.

- **Bitwise Mutation.** Flip each bit with probability $1/n$.
- **Random Selection.** Select one individual from $\Phi_t \cup \Psi_t$ at random.

Let $0, 1, 2$ represent the points $|x| = 0, 1, 2$ respectively. Transition probabilities of the $(1+1)$ EA among non-optimal points are given in Table VI.

TABLE VI
TRANSITION PROBABILITIES AMONG NON-OPTIMAL POINTS

$P(x,y)$	1	2
1	$\frac{5}{8}$	$\frac{1}{8}$
2	$\frac{1}{4}$	$\frac{2}{4}$

According to Theorem 1, the expected hitting time of the $(1+1)$ EA satisfies a linear equation system:

$$\begin{cases} \frac{3}{8}g^{(1+1)}(1) - \frac{1}{8}g^{(1+1)}(2) = 1, \\ -\frac{1}{4}g^{(1+1)}(1) + \frac{2}{4}g^{(1+1)}(2) = 1. \end{cases} \quad (51)$$

Solving the equations, we get the expected hitting time as follows:

$$g^{(1+1)}(1) = g^{(1+1)}(2) = 4. \quad (52)$$

Starting from any non-optimal point, the expected running time of the $(1+1)$ EA is 4.

Now we consider a simple $(2+2)$ EA which runs the two copies of the above $(1+1)$ EA independently. Let (i,j) represent the population (x_1, x_2) such that $|x_1| = i$, $|x_2| = j$ where $i, j = 0, 1, 2$. Transition probabilities of the (2+2) EA among non-optimal points are given in Table VII.

TABLE VII
TRANSITION PROBABILITIES AMONG NON-OPTIMAL POINTS

$P(X,Y)$	$(1,1)$	$(1,2)$	$(2,1)$	$(2,2)$
$(1,1)$	$\frac{5}{8}\cdot\frac{5}{8}$	$\frac{5}{8}\cdot\frac{1}{8}$	$\frac{1}{8}\cdot\frac{5}{8}$	$\frac{1}{8}\cdot\frac{1}{8}$
$(1,2)$	$\frac{5}{8}\cdot\frac{1}{4}$	$\frac{5}{8}\cdot\frac{2}{4}$	$\frac{1}{8}\cdot\frac{1}{4}$	$\frac{1}{8}\cdot\frac{2}{4}$
$(2,1)$	$\frac{1}{4}\cdot\frac{5}{8}$	$\frac{1}{4}\cdot\frac{1}{8}$	$\frac{2}{4}\cdot\frac{5}{8}$	$\frac{2}{4}\cdot\frac{1}{8}$
$(2,2)$	$\frac{1}{4}\cdot\frac{1}{4}$	$\frac{1}{4}\cdot\frac{2}{4}$	$\frac{2}{4}\cdot\frac{1}{4}$	$\frac{2}{4}\cdot\frac{2}{4}$

According to Theorem 1, the expected hitting time of the (2+2) EA satisfies a linear equation system:

$$\begin{cases} \frac{39}{64}g^{(2+2)}(1,1) - \frac{5}{64}g^{(2+2)}(1,2) \\ \quad -\frac{5}{64}g^{(2+2)}(2,1) - \frac{1}{64}g^{(2+2)}(2,2) = 1, \\ -\frac{5}{32}g^{(2+2)}(1,1) + \frac{22}{32}g^{(2+2)}(1,2) \\ \quad -\frac{1}{32}g^{(2+2)}(2,1) - \frac{2}{32}g^{(2+2)}(2,2) = 1, \\ -\frac{5}{32}g^{(2+2)}(1,1) - \frac{1}{32}g^{(2+2)}(1,2) \\ \quad +\frac{22}{32}g^{(2+2)}(2,1) - \frac{2}{32}g^{(2+2)}(2,2) = 1, \\ -\frac{1}{16}g^{(2+2)}(1,1) - \frac{2}{16}g^{(2+2)}(1,2) \\ \quad -\frac{2}{16}g^{(2+2)}(2,1) + \frac{12}{16}g^{(2+2)}(2,2) = 1. \end{cases} \quad (53)$$

Solving the equations, we get the expected hitting time as follows:

$$g^{(2+2)}(1,1)=g^{(2+2)}(1,2)=g^{(2+2)}(2,1)=g^{(2+2)}(2,2) = \frac{16}{7}. \quad (54)$$

Staring from any non-optimal point, the expected running time of the $(2+2)$ EA is $2 \times 16/7 = 32/7 \geq 4$. Therefore using a population doesn't shorten the expected running time on the deceptive function if the EA uses random selection. The reason is simple: the $(1+1)$ EA with random selection can accept a worse solution. This is a good strategy for deceptive functions. Hence using a population doesn't help too much.

VIII. CONCLUSIONS

This paper proposes population scalability for studying how population-size affects the computation time of population-based EAs. Population scalability is the ratio of the expected hitting time between a benchmark EA and an EA using a larger population size. Average drift analysis is presented as a

tool of comparing the expected hitting time of two EAs and estimating lower and upper bounds on population scalability.

Our results can be regarded as a rigorous analysis of several intuitive beliefs.

1) "Using a population may reduce the expected hitting time of an EA to find an optimal point." This belief is not always true. Two counter-examples are given, which are a $(1+\lambda)$ EA on two-paths-II functions and a $(\mu+1)$ EA on unimodal functions.
2) "Using a population cannot shorten the expected running time of an elitist EA on unimodal functions." This belief is always true for any $(\mu+\lambda)$ EAs with elitist selection on unimodal functions on the time-fitness landscape, but not always true in terms of the distance-based fitness landscape.
3) "Using a population can reduce the expected running time of an EA on deceptive functions." This belief is true under conditions. It is true for a $(\lambda+\lambda)$ EA if the benchmark $(1+1)$ EA uses elitist selection, but not true if the benchmark EA uses random selection.

More generally, for any fitness function f, there exist "good" $(1+1)$ elitist EAs to which f is unimodal [32]. Thus using a population is useless if a function is unimodal to the benchmark EA (or the benchmark EA is "good" to the function). In contrast, for any fitness function f, there exist "bad" $(1+1)$ elitist EAs to which f is deceptive [32]. Thus using a population may be useful if a function is deceptive to the benchmark EA (or the benchmark EA is "bad" to the function).

There still exist many open questions. For example, how to estimate the scalability threshold? how to analyse population scalability of EAs with crossover? what are practical criteria for judging population scalability? what is the optimal population size that maximises population scalability?

ACKNOWLEDGEMENT

The authors would like to thank Dr. Tianshi Chen for his comments on deceptive functions and the late Dr. Boris Mitavskiy for his comments on two-paths functions.

REFERENCES

[1] A. Prügel-Bennett, "Benefits of a population: five mechanisms that advantage population-based algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 500–517, 2010.
[2] D. Goldberg, K. Deb, and J. Clark, "Genetic algorithms, noise, and the sizing of populations," *Complex Systems*, vol. 6, pp. 333–362, 1992.
[3] H. Mühlenbein and D. Schlierkamp-Voosen, "The science of breeding and its application to the breeder genetic algorithm (BGA)," *Evolutionary Computation*, vol. 1, no. 4, pp. 335–360, 1993.
[4] J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS-a genetic algorithm with varying population size," in *Proceedings of 1994 IEEE World Congress on Computational Intelligence*. IEEE, 1994, pp. 73–78.
[5] A. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.
[6] G. Harik, E. Cantú-Paz, D. Goldberg, and B. Miller, "The gambler's ruin problem, genetic algorithms, and the sizing of populations," *Evolutionary Computation*, vol. 7, no. 3, pp. 231–253, 1999.
[7] J. He and X. Yao, "From an individual to a population: An analysis of the first hitting time of population-based evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 495–511, 2002.
[8] T. Jansen, K. de Jong, and I. Wegener, "On the choice of the offspring population size in evolutionary algorithms," *Evolutionary Computation*, vol. 13, no. 4, pp. 413–440, 2005.
[9] J. Jägersküpper and C. Witt, "Rigorous runtime analysis of a (μ+ 1) ES for the sphere function," in *Proceedings of 2005 Genetic and Evolutionary Computation Conference*, H.-G. Beyer and U.-M. O'Reilly, Eds. Washington DC, USA: ACM, 2005, pp. 849–856.
[10] C. Witt, "Population size versus runtime of a simple evolutionary algorithm," *Theoretical Computer Science*, vol. 403, no. 1, pp. 104–120, 2008.
[11] T. Storch, "On the choice of the parent population size," *Evolutionary Computation*, vol. 16, no. 4, pp. 557–578, 2008.
[12] Y. Yu and Z.-H. Zhou, "A new approach to estimating the expected first hitting time of evolutionary algorithms," *Artificial Intelligence*, vol. 172, no. 15, pp. 1809–1832, 2008.
[13] P. Oliveto, J. He, and X. Yao, "Analysis of population-based evolutionary algorithms for the vertex cover problem," in *Proceedings of 2008 IEEE Congress on Evolutionary Computation*. IEEE Press, 2008, pp. 1563–1570.
[14] T. Friedrich, P. Oliveto, D. Sudholt, and C. Witt, "Analysis of diversity-preserving mechanisms for global exploration," *Evolutionary Computation*, vol. 17, no. 4, pp. 455–476, 2009.
[15] T. Chen, J. He, G. Sun, G. Chen, and X. Yao, "A new approach for analyzing average time complexity of population-based evolutionary algorithms on unimodal problems," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 39, no. 5, pp. 1092–1106, 2009.
[16] J. Lässig and D. Sudholt, "Adaptive population models for offspring populations and parallel evolutionary algorithms," in *Proceedings of the 11th ACM/SIGEVO l Workshop on Foundations of Genetic Algorithms*, H.-G. Beyer and W. B. Langdon, Eds. Schwarzenberg, Austria: ACM, 2011, pp. 181–192.
[17] J. E. Rowe and D. Sudholt, "The choice of the offspring population size in the (1, λ) EA," in *Proceedings of 2012 Genetic and Evolutionary Computation Conference*, T. Soule and J. H. Moore, Eds. ACM, 2012, pp. 1349–1356.
[18] B. Doerr and M. Künnemann, "How the (1+ λ) evolutionary algorithm optimizes linear functions," in *Proceedings of 2013 Genetic and Evolutionary Computation Conference*. ACM, 2013, pp. 1589–1596.
[19] ——, "Royal road functions and the (1+ λ) evolutionary algorithm: Almost no speed-up from larger offspring populations," in *Proceedings of 2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 424–431.
[20] P. S. Oliveto and C. Witt, "On the runtime analysis of the simple genetic algorithm," *Theoretical Computer Science*, vol. 545, pp. 2–19, 2014.
[21] C. Gießen and C. Witt, "Population size vs. mutation strength for the (1+ λ) ea on onemax," in *Proceedings of 2015 Genetic and Evolutionary Computation Conference*. ACM, 2015, pp. 1439–1446.
[22] J. He and X. Yao, "Analysis of scalable parallel evolutionary algorithms," in *Proceedins of 2006 IEEE World Congress on Computational Intelligence*. Vancouver, Canada: IEEE Press, July 2006, pp. 427–434.
[23] ——, "Towards an analytic framework for analysing the computation time of evolutionary algorithms," *Artificial Intelligence*, vol. 145, no. 1-2, pp. 59–97, 2003.
[24] C. Grinstead and J. Snell, *Introduction to Probability*. American Mathematical Society, 1997.
[25] Y. Zhou and J. He, "A runtime analysis of evolutionary algorithms for constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 608–619, 2007.
[26] D. Andre and J. Koza, "A parallel implementation of genetic programming that achieves superlinear performance," *Information Sciences*, vol. 106, no. 3-4, pp. 201–218, 1998.
[27] E. Alba, "Parallel evolutionary algorithms can achieve superlinear performance," *Information Processing Letters*, vol. 82, no. 1, pp. 7–13, 2002.
[28] E. Alba and M. Tomassini, "Parallelism and evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 443–462, 2002.
[29] J. Jägersküpper, "A blend of markov-chain and drift analysis," in *Proceedings of 10th International Conference on Parallel Problem Solving from Nature*, G. Rudolph, T. Jansen, N. Beume, S. Lucas, and C. Poloni, Eds. Springer, 2008, pp. 41–51.
[30] B. Doerr, "Tutorial: Drift analysis," in *Proceedings of 2011 Genetic and Evolutionary Computation Conference*. ACM, 2011, pp. 1311–1320.
[31] J. He and X. Yao, "Drift analysis and average time complexity of evolutionary algorithms," *Artificial Intelligence*, vol. 127, no. 1, pp. 57–85, 2001.

[32] J. He, T. Chen, and X. Yao, "On the easiest and hardest fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 295–305, 2015.
[33] J. He, L. Kang, and Y. Chen, "Convergence of genetic evolution algorithms for optimization," *Parallel Algorithms and Applications*, vol. 5, no. 1, pp. 37–56, 1995.